Objectif calcul ouvert

Konrad Hinsen

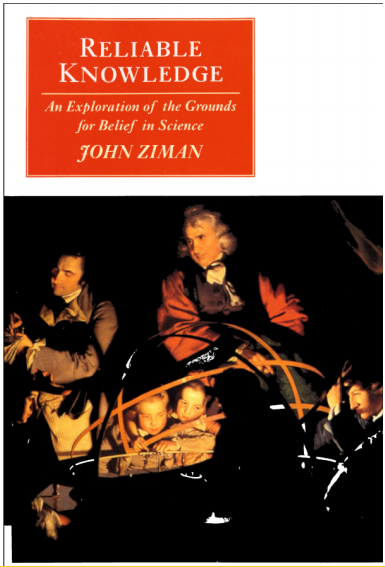
Centre de Biophysique Moléculaire, Orléans, France
and
Synchrotron SOLEIL, Saint Aubin, France

14 décembre 2022

Aperçu

La science ouverte

La science


RELIABLE
KNOWLEDGE
An Exploration of the Grounds for Belief in Science
JOHN ZIMAN

Fermé

Légalement ouvert

Effectivement ouvert

Une brève histoire de la science

1600 : Comprendre le monde

Science artisanale

- Chercheurs autonomes
- Rôles des institutions (universités, sociétés savantes) :
  - Communication
  - Formation
  - Respect des conventions

Une brève histoire de la science

1600 : Comprendre le monde

Science artisanale

1950 : Soutenir l'économie

Science industrielle

- Financement massif par les états
- Objectif : innovation
- Montée en puissance des institutions (universités, CNRS, ...)
- Productisation (publications, données, logiciels, ...)
- Gestion managériale :
  - Productivité
  - Compétitivité
  - Évaluation

Une brève histoire de la science

1600 : Comprendre le monde

Science artisanale

1950 : Soutenir l'économie

Science industrielle

2000 : Répondre aux défis sociétaux

Science sociétale

- Comprendre des phénomènes complexes (santé, climat, ...)
- Contribuer a l'émergence de consensus
- Intégrer la diversité des points de vue

La science ouverte

logique industrielle

logique sociétale

La science ouverte

Ouverture à l'inspection :

- Résultats : accès ouvert
- Provenance : données ouvertes, code ouvert
- Processus : évaluation ouverte, ...

La science ouverte

Ouverture à l'inspection :

- Résultats : accès ouvert
- Provenance : données ouvertes, code ouvert
- Processus : évaluation ouverte, ...

Ouverture à la participation :

- Diversité, inclusivité, ...
- "Science citoyenne"
- Infrastructures disponibles pour tous (Zenodo, ...)

Faire confiance à la science

Confiance : personnes et institutions

Hanzi Freinacht
@HFreinacht

Trust (def. from upcoming book):

1. competence (or credibility, or skill),
2. goodwill (i.e. you think they have good intentions),
3. reliability (they won't be flaky), and
4. alignment (that our interests align and don't contradict one another).

4:21 PM · Nov 18, 2022 · Twitter Web App

Confiance : produits, publications, logiciels, calculs

- Inspection, analyse
- Expérience d'utilisateur
- Processus de création
- Confiance dans les créateurs

Confiance par délégation

- Avis d'experts
- Réputation

Les bulles de confiance

Tout le monde dans mon domaine utilise logiciel X...

Les bulles de confiance

Tout le monde dans mon domaine utilise logiciel X...

... il est donc **digne de confiance** !

Une publication simple

5. ***Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen;***
von A. Einstein.

In dieser Arbeit soll gezeigt werden, daß nach der molekularkinetischen Theorie der Wärme in Flüssigkeiten suspendierte Körper von mikroskopisch sichtbarer Größe infolge der Molekularbewegung der Wärme Bewegungen von solcher Größe ausführen müssen, daß diese Bewegungen leicht mit dem Mikroskop nachgewiesen werden können. Es ist möglich, daß die hier zu behandelnden Bewegungen mit der sogenannten „Brownschen Molekularbewegung“ identisch sind; die mir erreichbaren Angaben über letztere sind jedoch so ungenau, daß ich mir hierüber kein Urteil bilden konnte.

Wenn sich die hier zu behandelnde Bewegung samt den für sie zu erwartenden Gesetzmäßigkeiten wirklich beobachten läßt, so ist die klassische Thermodynamik schon für mikroskopisch unterscheidbare Räume nicht mehr als genau gültig anzusehen und es ist dann eine exakte Bestimmung der wahren Atomgröße möglich. Erwiese sich umgekehrt die Voraussage dieser Bewegung als unzutreffend, so wäre damit ein schwerwiegendes Argument gegen die molekularkinetische Auffassung der Wärme gegeben.

A. Einstein, 1905

- un seul auteur
- raisonnement verbal, calculs niveau lycée
- accessible à tout étudiant en physique

Une publication complexe

Evolutionary-scale prediction of atomic level protein structure with a language model

Zeming Lin [1 2 *] **Halil Akin** [1 *] **Roshan Rao** [1 *] **Brian Hie** [1 3 *] **Zhongkai Zhu** [1] **Wenting Lu** [1] **Nikita Smetanin** [1] **Robert Verkuil** [1] **Ori Kabeli** [1] **Yaniv Shmueli** [1] **Allan dos Santos Costa** [4] **Maryam Fazel-Zarandi** [1] **Tom Sercu** [1 †] **Salvatore Candido** [1 †] **Alexander Rives** [1 † 2]

Abstract

Artificial intelligence has the potential to open insight into the structure of proteins at the scale of evolution. It has only recently been possible to extend protein structure prediction to two hundred million cataloged proteins. Characterizing the structures of the exponentially growing billions of protein sequences revealed by large scale gene sequencing experiments would necessitate a breakthrough in the speed of folding. Here we show that direct inference of structure from primary sequence using a large language model enables an order of magnitude speed-up in high resolution structure prediction. Leveraging the insight that language models learn evolutionary patterns across millions of sequences, we train models up to 15B parameters, the largest language model of proteins to date. As the language models are scaled they learn information that enables prediction of the three-dimensional structure of a protein at the resolution of individual atoms. This results

1. Introduction

The sequences of proteins at the scale of evolution contain an image of biological structure and function. This is because the biological properties of a protein act as constraints on the mutations to its sequence that are selected through evolution, recording structure and function into evolutionary patterns (1–3). Within a protein family, structure and function can be inferred from the patterns in sequences (4, 5). This insight has been central to progress in computational structure prediction starting from classical methods (6, 7), through the introduction of deep learning (8–11), up to the present state-of-the-art (12, 13).

The idea that biological structure and function are reflected in the patterns of protein sequences has also motivated a new line of research on evolutionary scale language models (14). Beginning with Shannon's model for the entropy of text (15), language models of increasing complexity have been developed to fit the statistics of text, culminating in modern large-scale attention based architectures (16–18). Language models trained on the amino acid sequences of millions of diverse proteins have the potential to learn patterns across

Z. Lin et al., 2022

- 15 auteurs
- modèle avec 15 milliards de paramètres
- code et données disponibles...
- ... mais difficiles à utiliser
- exécution en ligne proposée...
- ... mais qui sait quel code y tourne ?

Une publication complexe

Evolutionary-scale prediction of atomic level protein structure with a language model

Zeming Lin [1 2 *] **Halil Akin** [1 *] **Roshan Rao** [1 *] **Brian Hie** [1 3 *] **Zhongkai Zhu** [1] **Wenting Lu** [1] **Nikita Smetanin** [1] **Robert Verkuil** [1] **Ori Kabeli** [1] **Yaniv Shmueli** [1] **Allan dos Santos Costa** [4] **Maryam Fazel-Zarandi** [1] **Tom Sercu** [1 †] **Salvatore Candido** [1 †] **Alexander Rives** [1 † 2]

Abstract

Artificial intelligence has the potential to open insight into the structure of proteins at the scale of evolution. It has only recently been possible to extend protein structure prediction to two hundred million cataloged proteins. Characterizing the structures of the exponentially growing billions of protein sequences revealed by large scale gene sequencing experiments would necessitate a breakthrough in the speed of folding. Here we show that direct inference of structure from primary sequence using a large language model enables an order of magnitude speed-up in high resolution structure prediction. Leveraging the insight that language models learn evolutionary patterns across millions of sequences, we train models up to 15B parameters, the largest language model of proteins to date. As the language models are scaled they learn information that enables prediction of the three-dimensional structure of a protein at the resolution of individual atoms. This results

1. Introduction

The sequences of proteins at the scale of evolution contain an image of biological structure and function. This is because the biological properties of a protein act as constraints on the mutations to its sequence that are selected through evolution, recording structure and function into evolutionary patterns (1–3). Within a protein family, structure and function can be inferred from the patterns in sequences (4, 5). This insight has been central to progress in computational structure prediction starting from classical methods (6, 7), through the introduction of deep learning (8–11), up to the present state-of-the-art (12, 13).

The idea that biological structure and function are reflected in the patterns of protein sequences has also motivated a new line of research on evolutionary scale language models (14). Beginning with Shannon's model for the entropy of text (15), language models of increasing complexity have been developed to fit the statistics of text, culminating in modern large-scale attention based architectures (16–18). Language models trained on the amino acid sequences of millions of diverse proteins have the potential to learn patterns across

Z. Lin et al., 2022

- 15 auteurs
- modèle avec 15 milliards de paramètres
- code et données disponibles...
- ... mais difficiles à utiliser
- exécution en ligne proposée...
- ... mais qui sait quel code y tourne ?

Légalement ouvert,
pas effectivement ouvert

Un logiciel simple

COMPUTER PHYSICS COMMUNICATIONS 1 (1969) 21–24. NORTH-HOLLAND PUBLISHING COMP., AMSTERDAM

A PROGRAM TO CALCULATE FRANCK-CONDON FACTORS

A. C. ALLISON
Harvard College Observatory and Smithsonian Astrophysical Observatory, Cambridge, Massachusetts 02138, *USA*

Received 18 March 1969

PROGRAM SUMMARY

Title of program (32 characters maximum): FRANK-CONDON FACTOR PROGRAM

Catalogue number: AACA

Computer for which the program is designed and others upon which it is operable

Computer: CDC 6400; ANY CDC 6000 SERIES. *Installation*: Smithsonian Astrophysical Obs., Cambridge, Mass., USA

Operating system or monitor under which the program is executed: SCOPE

Programming languages used: FORTRAN

High speed store required: 15,600 words. *No. of bits in a word*: 60

Is the program overlaid? No

No. of magnetic tapes required: None

What other peripherals are used? Card Reader; Line Printer

No. of cards in combined program and test deck: 537

Card punching code: C.D.C.

Keywords descriptive of problem and method of solution: Atomic, Structure, Transition, Franck-Condon, Bound States, Eigenvalues, Eigenfunctions, Schrödinger eq., Potential, Local, Numerov.

Un logiciel complexe

SOFTWARE ENGINEERING FOR CSE

Streamlining Development of a Multimillion-Line Computational Chemistry Code

Robin M. Betz and Ross C. Walker | San Diego Supercomputer Center

Software engineering methodologies can be helpful in computational science and engineering projects. Here, a continuous integration software engineering strategy is applied to a multimillion-line molecular dynamics code; the implementation both streamlines the development and release process and unifies a team of widely distributed, academic developers.

Un logiciel complexe et payant (mais : code source disponible)

Welcome to Amber!

Amber is a suite of biomolecular simulation programs. It began in the late 1970's, and is maintained by an active development community; see our history page and our contributors page for more information.

The term "Amber" refers to two things. First, it is a set of molecular mechanical force fields for the simulation of biomolecules (these force fields are in the public domain, and are used in a variety of simulation programs). Second, it is a package of molecular simulation programs which includes source code and demos.

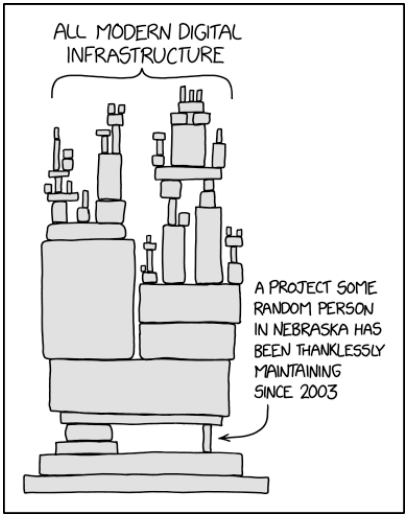
Amber is distributed in two parts: AmberTools22 and Amber22. You can use AmberTools22 without Amber22, but not *vice versa*. See the Download Amber link for information on how to download the code.

How to obtain Amber22

Amber22 facilitates faster simulations (on parallel CPU or GPU hardware) and is distributed with a separate license and fee structure. Click here for the Amber 22 License Agreement (PDF). Print this form, fill it out, sign and return (with your payment) to the address given at the bottom of the license agreement. Once your order is processed, you will receive download information via email. PDF versions of the Reference Manual are included in the download.

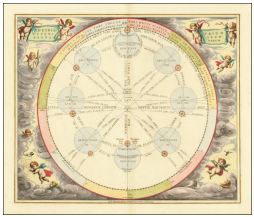
Sur les épaules de géants

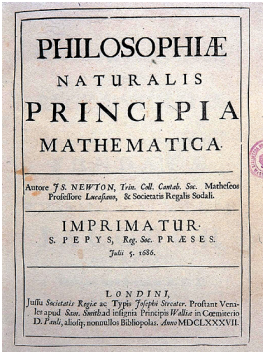
Sur des supports fragiles



https ://xkcd.com/2347

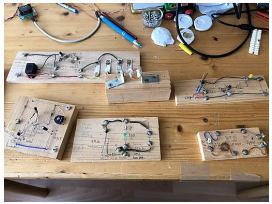
La consolidation de la connaissance

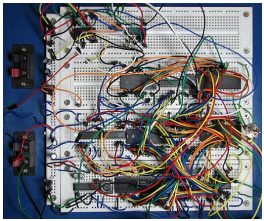


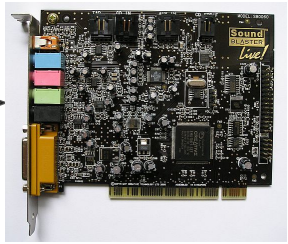


empirique → explicatif → computationnel

La consolidation technologique







bricolage → artisanat → ingénierie → industrie

Rendre la confiance explicite

- Nous devons **évaluer la fiabilité** des supports de nos recherches...
- ... et les **discuter** dans nos publications !

Le calcul ouvert

Le calcul ouvert

- permet **et facilite** l'examination critique
- **inspire** confiance

Le calcul ouvert

- permet **et facilite** l'examination critique
- **inspire** confiance

Le défi : passer du **légalement** ouvert à l'**effectivement** ouvert

Le calcul ouvert

Calcul ouvert =

code ouvert + **données** ouvertes

Le calcul ouvert

Calcul ouvert =

code ouvert + **données** ouvertes

Code ouvert =

logiciel ouvert + **environnement** ouvert

Le calcul ouvert

Calcul ouvert =

code ouvert + **données** ouvertes

Code ouvert =

logiciel ouvert + **environnement** ouvert

Logiciel : le code qui m'intéresse

Environnement : le code que ne m'intéresse pas

Le code ouvert

- permet **et facilite** l'examination critique
- **il ne suffit pas** de rendre le code public

Le code ouvert

- permet **et facilite** l'examination critique
- **il ne suffit pas** de rendre le code public
- il faut ouvrir **le logiciel et l'environnement**

Le code ouvert

- permet **et facilite** l'examination critique
- **il ne suffit pas** de rendre le code public
- il faut ouvrir **le logiciel et l'environnement**
- il faut **évaluer** la **fiabilité** de ses dépendances

Faire confiance aux logiciels

Le logiciel industriel

- Développé par des professionnels
- Spécification et documentation pour les utilisateurs
- Stable, bien testé, ...
- A fait ses preuves

Exemples : gcc, BLAS, Debian, Coq, scikit-lean

Faire confiance aux logiciels

Le logiciel industriel

- Développé par des professionnels
- Spécification et documentation pour les utilisateurs
- Stable, bien testé, ...
- A fait ses preuves

Exemples : gcc, BLAS, Debian, Coq, scikit-lean

Le logiciel artisanal

- Simple, compact
- Vérifiable par les pairs
- Ne dépend que de logiciels industriels

Exemples : les codes Fortran des années 1960 à 1990

Faire confiance aux logiciels industriels

Comme pour un médicament, un TGV...

Faire confiance aux logiciels industriels

Comme pour un médicament, un TGV...

Sources de confiance :

- Technologies éprouvées

Faire confiance aux logiciels industriels

Comme pour un médicament, un TGV...

Sources de confiance :

- Technologies éprouvées
- Évaluation par des experts indépendants

Faire confiance aux logiciels industriels

Comme pour un médicament, un TGV...

Sources de confiance :

- Technologies éprouvées
- Évaluation par des experts indépendants
- Modes d'emploi, formations, ...

Faire confiance aux logiciels industriels

Comme pour un médicament, un TGV...

Sources de confiance :

- Technologies éprouvées
- Évaluation par des experts indépendants
- Modes d'emploi, formations, ...
- Labels de qualité, marques de certification

Faire confiance aux logiciels industriels

Comme pour un médicament, un TGV...

Sources de confiance :

- Technologies éprouvées
- Évaluation par des experts indépendants
- Modes d'emploi, formations, ...
- Labels de qualité, marques de certification
- Cadre légal ou conventionnel de bonnes pratiques

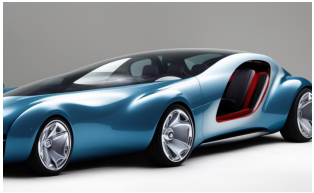
Faire confiance aux logiciels industriels

Comme pour un médicament, un TGV...

Sources de confiance :

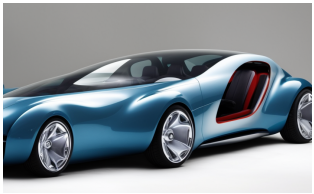
- Technologies éprouvées
- Évaluation par des experts indépendants
- Modes d'emploi, formations, ...
- Labels de qualité, marques de certification
- Cadre légal ou conventionnel de bonnes pratiques

Les prototypes



(Stable Diffusion 2)

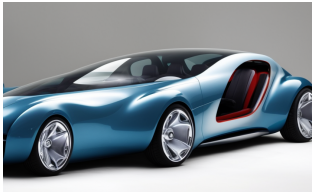
Les prototypes



(Stable Diffusion 2)

(Cjp24, CC BY-SA 4.0, via Wikimedia Commons)

Les prototypes



(Stable Diffusion 2)

(Cjp24, CC BY-SA 4.0, via Wikimedia Commons)

julia

Les logiciels pseudo-industriels

- se présentent comme des bibliothèques réutilisables
- trop grands pour une évaluation à l'artisanale
- trop petits pour une gestion à l'industrielle
  - trop spécialisés
  - évolution trop rapide

Un logiciel pseudo-industriel

The Molecular Modeling Toolkit

The Molecular Modelling Toolkit (MMTK) is an Open Source program library for molecular simulation applications. In addition to providing ready-to-use implementations of standard algorithms, MMTK serves as a code basis that can be easily extended and modified to deal with standard and non-standard problems in molecular simulations.

The three most common usage patterns of MMTK are:

- Writing Python scripts that make use of MMTK functions for standard simulation and modelling applications. This is similar to using other simulation packages with a scripting language (i.e. CHARMM or Gromos), but with the added advantage of having access to lots of useful Python modules from elsewhere.
- Writing modules that interact closely with MMTK (and perhaps other packages) to solve problems for which no standard solution exists. For example, adding a particular force field term or a particular simulation or analysis algorithm. There is not much competition for MMTK in that domain.
- Writing application programs in Python that use MMTK internally, for users that do not need to know anything about such internals. Those programs can provide easy-to-use graphical interfaces (see e.g. DomainFinder and nMOLDYN) or be integrated into a Web service (see e.g. WEBnm@).

K. Hinsen, 1997 - 2020

Pourquoi tant de logiciels pseudo-industriels dans la recherche ?

- le “vrai” industriel est trop contraignant et trop coûteux
- le code réutilisable fait partie des bonnes pratiques

Pourquoi tant de logiciels pseudo-industriels dans la recherche ?

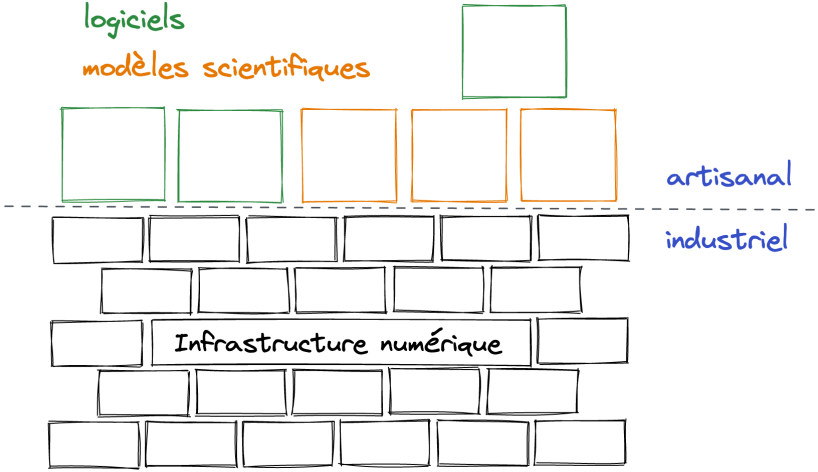
- le “vrai” industriel est trop contraignant et trop coûteux
- le code réutilisable fait partie des bonnes pratiques

Donald Knuth en 2008, dans un entretien avec Andrew Binstock :

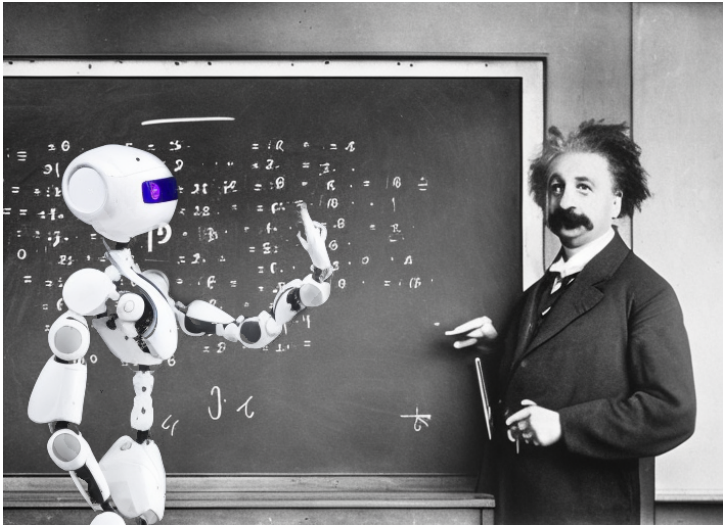
> *I also must confess to a strong bias against the fashion for reusable code. To me, "re-editable code" is much, much better than an untouchable black box or toolkit. I could go on and on about this. If you’re totally convinced that reusable code is wonderful, I probably won’t be able to sway you anyway, but you’ll never convince me that reusable code isn’t mostly a menace.*

https://www.informit.com/articles/article.aspx?p=1193856

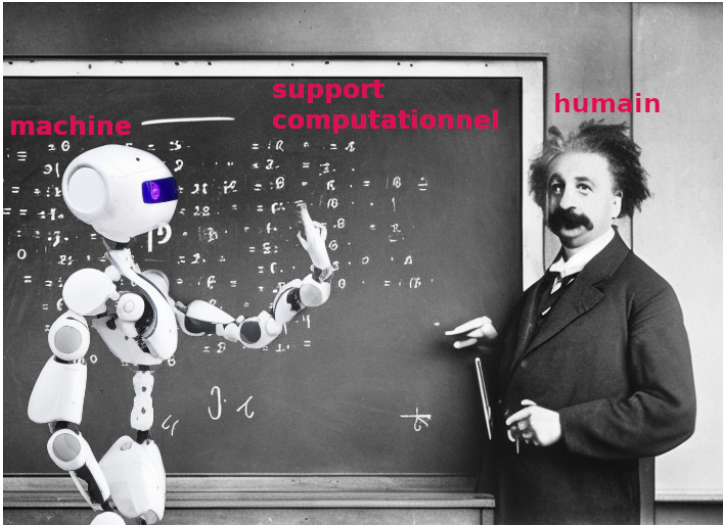
Restructuration



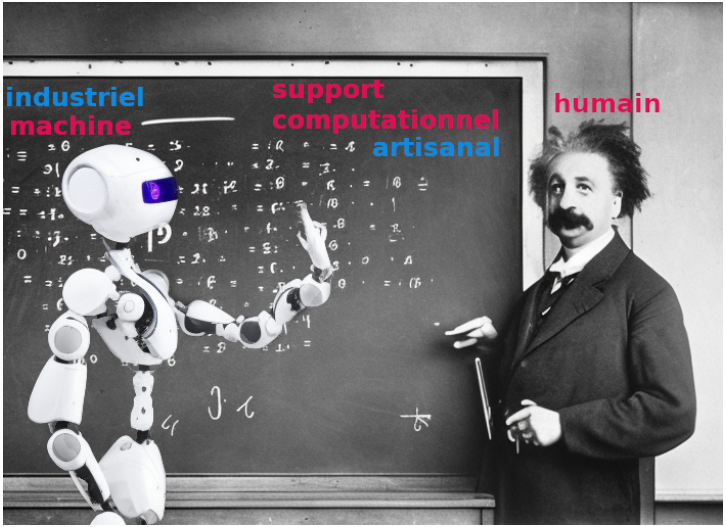
Le calcul effectivement ouvert



Le calcul effectivement ouvert



Le calcul effectivement ouvert



Supports computationnels (computational media)

Supports numériques pouvant encoder des calculs.

Exemples :

- langages de programmation
- tableurs
- pages Web
- moteurs de jeux
- notebooks

Agenda

1. Construire une infrastructure numérique **pour la recherche**
2. Développer des bonnes pratiques du logiciel **pour la recherche**
3. Séparer la science artisanale des outils industriels

Infrastructure numérique pour la recherche

Infrastructure technique

- Matériel, logiciels système, réseaux
- Modèles de données, formats de données, outils de partage et d'archivage
- Langages et outils de programmation
- Bibliothèques scientifiques d'intérêt général

Infrastructure numérique pour la recherche

Infrastructure technique

- Matériel, logiciels système, réseaux
- Modèles de données, formats de données, outils de partage et d'archivage
- Langages et outils de programmation
- Bibliothèques scientifiques d'intérêt géneral

Infrastructure instutionnelle

- Institutions dédiées au développement et à l'évaluation
- Gouvernance impliquant les utilisateurs

De retour dans votre laboratoire...

Posez-vous trois questions :

1. Pourquoi fais-je confiance à mes calculs ?
2. Pourquoi fais-je confiance aux calculs des autres ?
3. Quelle est la fiabilité de mes outils logiciels ?